

A SIGNATURE BASED DIGITAL IC TESTER

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

By
R N SHAH

to the
DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
AUGUST, 1980

10 11 7 8 80
m

CERTIFICATE

Certified that this work "A Signature Based Digital
IC. Tester" by R.N. Shah is carried out under my supervision
and is not submitted elsewhere for a degree.

August 6 ,1980

R.N. Biswas
(R.N. BISWAS)
Professor
Department of Electrical Engineering
Indian Institute of Technology
KANPUR

IT KANPUR
TRAL LIBRARY

No. 63796

0 NOV 1980

EE-1980-M-SHA-SIG

ACKNOWLEDGEMENT

I sincerely thank Dr. R.N. Biswas for his valuable guidance, suggestions and help given for successful completion of this work.

I express sincere thanks to all my friends who helped me. My thanks also go to the staff of I E. Deptt., particularly Shri S N Sikdar, for their kind co-operation

I also thank Shri C.M. Abraham for his neat and efficient typing work

Rajen N. Shah

TABLE OF CONTENTS

	Page No.
CHAPTER 1 INTRODUCTION	1
CHAPTER 2 THEORY OF SIGNATURE TESTING	4
2.1 Binary-Weighted Repetition Method	
2.2 Probabilistic Testing Methods	
CHAPTER 3 GENERAL SYSTEM CONSIDERATIONS	16
3.1 Test Pattern for Combinational Circuits	
3.2 Input Groups for Sequential Circuits	
3.3 Typical Test Cycle	
3.4 System Block Diagram	
CHAPTER 4 TEST SIGNAL GENERATOR AND CONTROL	23
4.1 Clock Generator and Speed Selector	
4.2 Test Pattern Generator	
4.3 Asynchronous Signal Generator	
4.4 Clock Bursts	
4.5 Control Circuit	
4.6 Delay Compensation	
CHAPTER 5 INPUT SIGNAL SELECTION	35
5.1 Pin Data Register and Data Loading Circuitry	
5.2 Signal Multiplexers	
5.3 PIN Decoder	

CHAPTER 6	INTERFACE CIRCUIT AND SIGNATURE COUNTER	41
6.1	Interface Circuit	
6.2	Analog multiplexer and Bad Level Detector	
6.3	Signature Counter and Display	
CHAPTER 7	CONCLUSION	51
7.1	Operating Instructions	
7.2	Suggested Future Development	
	BIBLIOGRAPHY	57
	APPENDIX I	58

LIST OF FIGURES

Figure		Page
3.1	System block diagram	21
4.1	Block diagram of control and signal generation circuits	32
4.2	Control and signal generation circuitry	33
4.3	IC layout on PCB	34
5.1	Block diagram of signal selector, pin address register and switch decoding	38
5.2	Signal selectro, pin address register and switch decoders circuitry	39
5.3	IC layout on PCB	40
6.1	Block diagram of signal conditioner, loading circuitry and signature counter	42
6.2	Schematic of various switches and loading circuitry	43
6.3	Loading circuitry	43
6 4	Level translator, loading circuitry and signature tester	50

LIST OF TABLES

Table	Page
2.1.1 2^N functions for n-input variables	6
2.1.2 Time for one test cycle	8
2.2.1 Values of a_j	10
2.2.2 Values of T and M for different n	14
3.2.1 Pin numbers and signals available for selection	19
6.1.1 Interface switch positions	44
7.1.1 Codes and test results of 4 input nand gate	53
7.1.2 Codes and test results of 74191 counter	55

ABSTRACT

An attempt has been made to design a digital IC tester based on the theory of signature testing. The signature chosen is the number of ones in the output sequence of the IC under test in response to an input test pattern. Two kinds of patterns are generated (1) Binary-Weighted Repetition Sequences, where the possible 2^n input combinations of n input variables are applied according to a pattern with the $(k+1)^{\text{th}}$ input combination repeated 2^k times ($k = 0, 1, \dots, 2^n$), (ii) Pseudo Random Binary Sequences, where all possible input combinations are applied in a pseudo-random fashion.

It has been shown that the Binary-Weighted Repetition Sequences, as an input test pattern, would test the logic of circuits satisfactorily and would detect a large number of faults. Pseudo random binary sequences are used for high-frequency testing.

Test procedure consists of feeding the information about various pins of the IC under test, in the form of certain code and comparing the output signature with its predetermined value. ICs can be tested at various frequencies (0.5 MHz to 10 MHz) for specified noise margin and propagation delays, under programmable load conditions, with worst-case input voltage levels.

CHAPTER 1

INTRODUCTION

Design of complex digital systems has been rendered quite simple with the advent of compatible logic families like TTL and CMOS. As all system designs are based on the logic and the worst-case specifications of the IC (Integrated Circuit) chips employed, it is a matter of great interest to test the logic of these ICs under worst-case conditions

Three types of testing, d-c (for functional specifications), a-c (for parametric specifications) and speed testing (for transient and clock-rate specifications) are usually performed on ICs. In d-c- testing binary patterns are applied as inputs to the IC under test, and its corresponding steady-state outputs are analysed to determine the correct functional behaviour. A-C test involves the measurement of actual voltage and current levels, whereas speed testing is the same as d-c- testing, but performed with inputs switching at or near maximum specified speed

Ideally one should like to perform all the three kinds of tests. Thus an IC tester should test the specified logic at the highest rated frequency for the specified noise margin and propagation delay, under maximum load conditions, with worst-case input and supply voltage levels.

One of the methods of testing is to apply identical pseudo-random binary sequences (PRBS) simultaneously to the device under test (DUT) and to a reference units (good replica of DUT) and then, to compare the output sequences generated by both of them, any discrepancy is said to constitute an error or fault in DUT. This method is essentially restricted to d-c-testing, as the failure limits of the DUT can never be actually determined. This limitation, coupled with the need of a reference device having guaranteed specifications, render this method rather inconvenient.

The other approach emulates the DUT on a computer. The test programs are automatically translated to the appropriate input stimuli and outputs are automatically processed and interpreted by the computer. Here the speed of testing is limited by the processor speed. Moreover, emulation of the large number of existing ICs is also a difficult task. The user has to write a program to emulate the DUT each time he wants to test a new device.

The third method is signature testing. The signature of the DUT output, in response to a PRBS input, is compared with the already known signature of an identical good device. The signature in general can be any function of the output waveform, the simplest choices being

- 1) LEVEL signature, given by the duration for which the output is at one of the two logic levels in one complete test sequence.

- 11) TRANSITION signature, given by the number of transitions that the output makes in one complete test sequence.

Such simple signatures are readily detected by means of simple counters. No modelling and associated computer test generation and/or emulation are required. Hence the DUT can be tested at its maximum rate. The reference signature can be obtained once for all with a functionally good device tested at a lower clock-rate.

In this project an attempt has been made to build an IC tester based on LEVEL signature testing. The basic theory behind signature testing is discussed in the second chapter. Chapter three deals with the adaptation of this theory to practical IC testing and gives the information about the kind of signals generated. Chapters 4,5 and 6 give detailed design of the circuitry followed by the specifications, operating instructions and possible future development of the instrument in Chapter 7.

CHAPTER 2

THEORY OF SIGNATURE TESTING

Since the burgeoning growth of sophisticated digital systems from the 1960's, a lot of interest has been created in the subjects of testing, maintenance and reliability of digital systems. Several studies have been carried out attempting to predict analytically the effectiveness of applying random input patterns for fault detection and diagnoses. While such analytic predictions have been possible in the case of combinational circuits, the existence of feedback loops and initial states have not permitted sequential circuits to be so analysed. In principle, any combinational circuit can be tested deterministically, without going for PRBS inputs. We consider the merits and demerits of such a scheme first and then go to the probabilistic or random testing.

2.1 BINARY WEIGHTED REPETITION METHOD

Consider a combinational circuit with n inputs. The problem of fault detection can be looked upon as the problem of distinguishing between the true logic function specified for the circuit and any of the other possible logic functions of n variables as any one of these may be the result of a fault. Evidently this distinction will be perceptible, at a given output, only for some of the possible input combinations. Hence

for a general-purpose test system, where the true logic as well as the fault logic can be any two of all possible logic functions of the circuit under test, one must evolve a scheme where the selected signature is distinct for every possible logic function.

Let us represent each combination of the n input variables as an n -dimensional input vector, denoted by $A_k, k=0, 1, 2, \dots, N-1$, where $N = 2^n$. A logic function is defined by the values (0 or 1) it produces at the output for the N distinct input vectors. As such, there are 2^N possible logic functions, implying thereby that for every true logic function, there are $2^N - 1$ likely logic functions, any one of which can be the result of a fault. Let us denote the logic functions by $F_i, i = 0, 1, 2, \dots, 2^N - 1$, and the output due to F_i in response to an input vector A_k by y_{ik} . For the sake of convenience, let us order the A_k 's and F_i 's in the following manner.

$$A_k = (k_{N-1}, k_{N-2}, \dots, k_1, k_0),$$

where

$$k = \sum_{j=0}^{N-1} k_j 2^j \quad (2.1.1)$$

i.e. A_k represents the number k in binary code, and

$$i = \sum_{k=0}^{N-1} y_{ik} 2^k, \quad (2.1.2)$$

Let us consider a test procedure in which different input vectors A_0, A_1, \dots, A_{N-1} are applied sequentially, with A_k repeated m_k times, and let the signature be defined to be the number of steps in the input sequence for which the output is at the '1' level. Then the value of the signature for the logic F_1 will be given by

$$L_1 = \sum_{k=0}^{N-1} y_{1k} m_k \quad (2.1.3)$$

As y_{1k} 's are either '0' or '1', all the possible values of L_1 for a chosen set of m_k 's form a set of numbers obtained by all possible selective sums of m_0, m_1, \dots, m_{N-1} . i.e.,

$$L_i = \sum_{\substack{k \\ (y_{1k}=1)}} m_k \quad (2.1.4)$$

e.g. for the case $n = 2$ we get from Table 2.1.1. $L_0 = 0$, $L_1 = m_0$, $L_2 = m_0 + m_1$. In order that any two functions have distinct signatures we need

$$L_i \neq L_j \quad \text{for} \quad j \neq i \\ i, j = 0, 1, 2, \dots, 2^{N-1} \quad (2.1.5)$$

It is well known that the binary weighted scheme i.e.

$$m_k = 2^k, \quad (2.1.6)$$

satisfies 2.1.5, hence this choice can be adopted for deterministic signature method. We call this method the Binary Weighted Repetition (BWR) method.

The total number of input steps needed for one such complete test cycle is given by

$$T = \sum_{k=0}^{N-1} m_k = 2^N - 1 \quad (2.1.7)$$

Table 2.1.2 gives the time required for one complete test cycle for different values of n . The input vectors are applied at the rate of 10 MHz

Table 2 1.2 Time for one test cycle

n	N	T	time @ 10 MHz
2	4	15	1.5 μ sec
3	8	255	25.5 μ sec
4	16	65535	6.5535 msec
5	32	4.3×10^9	430 seconds
6	64	1.8×10^{19}	6×10^4 years

One can readily see that this method of testing is quite impracticable beyond 5 input variables. Even with 5 inputs one needs quite a long time (>7 minutes) and a 32 bit counter to obtain the level signature. Thus deterministic testing of

combinational circuits is really possible for only upto 4 inputs.

2.2 PROBABILISTIC TESTING METHODS

Let us again consider an n-input combinational circuit, but now with n random inputs. The level signature would now become, the probability of getting a '1' at the output. And the level probability signature is given by

$$L_i = \sum_{k=0}^{N-1} y_{ik} p_k \quad (2.2.1)$$

where p_k is the probability that an input vector would be A_k . This expression is similar to the eqn. 2.1.3 with m_k replaced by the probability p_k . The same binary weightage should, therefore, now be applied to the probabilities, and as

$$\sum_{k=0}^{N-1} p_k = 1$$

we have

$$p_k = 2^k / \sum_{k=0}^{N-1} 2^k = 2^k / (2^N - 1) \quad (2.2.2)$$

A suitable PRBS test sequence for the input vectors has to be selected to satisfy eqn. 2.2.2, in order to design a testing scheme which will ensure distinct values of the level probability signature L_i for different fault modes. For such a scheme, the

values of L_1 's can be determined using eqns. 2.1.2, 2.2.1 and 2.2.2.

$$L_1 = \frac{1}{2^N - 1} \quad i = 0, 1, \dots, 2^N - 1 \quad (2.2.3)$$

Thus 2^N different values exist for the level probability signature L_1 corresponding to 2^N logic functions.

McCluskey [1] has shown that eqn. 2.2.2 is satisfied, if the probability of the i th individual input to be a '1' is given by

$$a_j = 1 - \frac{1}{2^{2^j} + 1} \quad j = 0, 1, \dots \quad (2.2.4)$$

A simple proof of this result is given in Appendix I. Table 2.2.1 gives the values of a_j for a few values of j

Table 2.2.1 Values of a_j

j	1	2	3	4	5
a_j	1/3	1/5	1/17	1/129	1/66537

A glance at this table is sufficient to indicate the impracticability of realising a PRBS with such values of a_j .

Losq [2] has proposed an alternative scheme using equal probabilities for all individual values, i.e.,

$$a_j = a \quad \text{for all } j \quad (2.2.5)$$

Under this simple condition, the probability for an input vector A_k is given by

$$p_k = a^l (1-a)^{n-l} \quad \begin{matrix} l = 0 \dots n \\ k = 0 \dots N-1 \end{matrix} \quad (2.2.6)$$

where l is the number of '1's in the binary representation of k . Let the input vectors having a particular value of l , be said to constitute an input vector group. An input vector group for a given value of l consists of nC_l input vectors with the same value of p_k . The number of distinct expressions for p_k thus reduces from 2^n to $n+1$, the possible values of l being $0, 1, \dots, n$. However even though p_k is a different function of 'a' for different values of l resulting values of p_k may not all be distinct.

The following analysis shows, the test length T increases very fast as n increases, if one has to ensure that all logic functions distinguishable by equal-probability testing actually have distinct values of the signature.

In general, 'a' can be expressed as the ratio of two relatively prime integers.

$$a = r/s \quad (2.2.7)$$

where r and s are integers and $r < s$. Then

$$p_k = r^l \frac{(s-r)^{n-l}}{s^n} \quad (2.2.8)$$

where A_k has 1 '1's and $n-1$ '0's

This can be realized only if each input vector A_k is repeated m_1 times in a test cycle, where

$$m_1 = s^1(s-r)^{n-1} \quad (2.2.9)$$

and the test length is given by

$$T = s^n \quad (2.2.10)$$

For the simplest case $a = \frac{1}{2}$, this gives

$$m_1 = 1 \quad \text{and} \quad T = 2^n$$

and for $a = 2/3$ gives

$$m_1 = 2^1 \quad \text{and} \quad T = 2^n$$

Thus $a = 2/3$ corresponds to the binary weighted equal probabilities.

The upper bound of distinguishable logic functions is given by the number M of distinct values taken by all possible selective sums of m_1 . As each m_1 can be repeated upto m_1 times in such a selective sum, there are ${}^n C_1 + 1$ ways of choosing terms from an input vector group 1. M can, therefore, be expressed as a continued product as follows .

$$M = \prod_{l=0}^n ({}^n C_l + 1) \quad (2.2.11)$$

Let us next determine the minimum value of s required so as to make M realisable

The problem is to ensure that a combination of the form $u m_1 + v m_{1+j}$, does not have the same numerical value for two sets of choices of u for a given choice of l and j , i.e.

$$u_1 m_1 + v_1 m_{1+j} \neq u_2 m_1 + v_2 m_{1+j} \quad (2.2.12)$$

where $u_2 \neq u_1$ and $v_2 \neq v_1$

This implies that

$$\frac{m_{1+j}}{m_1} \neq \frac{u_1 - u_2}{v_2 - v_1} \quad (2.2.13)$$

Using eqn. 2.2.9 this can be written as

$$\left(\frac{r}{s-r}\right)^j \neq \frac{u_1 - u_2}{v_2 - v_1} \quad (2.2.14)$$

Inequality 2.2.14 can be ensured for arbitrary values of j , u_1 , u_2 , v_1 and v_2 , if

$$\left(\frac{r}{s-r}\right)^{\pm 1} > nC_{1 \max} \quad , \quad (2.2.15)$$

the \pm sign corresponding to the two complementary situations $a > \frac{1}{2}$ ($r > s-r$) and $a < \frac{1}{2}$ ($r < s-r$) respectively. Let us without loss of generality, choose $a > \frac{1}{2}$. Then we have

$$\frac{s}{r} > 1 + nC_{1 \max} \quad (2.2.16)$$

The choice $r = 1$ leads to the minimum test length

$$T_{\min} = (2 + nC_{1 \max})^n \quad , \quad (2.2.17)$$

Corresponding to a probability

$$a = \frac{1}{2 + {}^nC_1_{\max}} \quad (2.2.18)$$

The required values of m_1 are given by eqn. 2.2.9 as

$$m_1 = (2 + {}^nC_1_{\max})^{n-1} \quad (2.2.19)$$

The Table 2.2.2 gives the values of T, M and required 'a' for different values of n.

Table 2.2.2 Values of T and M for different n

n	$T_{\text{BWRs}} = 2^N$	T	M	a
2	16	12	16	3/4
4	2^{16}	700	2^{12}	7/8
6	2^{64}	10^6	1.1×10^8	21/22
8	2^{256}	4.4×10^{12}	7.2×10^{14}	71/72

From this table it can be readily seen that it is not practicable to generate equal probability sequences, which would distinguish all possible distinguishable logic functions for $n > 6$.

Losq[2] has derived the following expression for the probability p for a faulty device to be accepted or for a good device be rejected

$$P = \frac{2}{\pi} \frac{1}{\sqrt{2 \pi \sigma^2}} \exp \left(- \left(\frac{L_1 - .5}{2} \right)^2 \right) \quad (2.2.20)$$

where

σ is variance given by

$$\sigma = (a^2 + (1-a^2))^n / 4,$$

and the other symbols have the same interpretation as before.

It may be noted that the choice given by eqn. 2.2.9 is not in any way unique. Other choices of m_1 may also lead to meaningful testing schemes, e.g. the choice

$$m_k = k + 1 \quad (2.2.21)$$

is capable of distinguishing $\frac{1}{2} N (N+1)$ different functions, with a test length of $\frac{1}{2} N (N+1)$

CHAPTER 3

GENERAL SYSTEM CONSIDERATION

In this project test methods are developed for 16-pin ICs. However the same principle, with a little modification, could be extended for larger ICs. Like any digital circuit, an IC can be a combinational circuit or a sequential circuit, leading to different input requirements. The first section of this chapter discusses the test patterns used for combinational circuits, followed by the input requirements of sequential circuits. In order to develop a general test procedure for all 16-pin ICs, the input requirements of various ICs have been tabulated. The next section describes a typical test cycle for any IC and the chapter is concluded with the system block diagram.

3.1 TEST PATTERNS FOR COMBINATIONAL CIRCUITS

The two possibilities of test patterns discussed in the previous chapter are

- 1) BWRS,
- 11) Equal probability PRBS

As pointed out in Section 2.1, BWRS is impracticable for number of inputs $n \geq 4$. For $n = 4$ a BWRS test pattern distinguisher between 2^{16} different logic functions. Even if these four outputs of BWRS are repeated, the complete test cycle would still

distinguish 2^{16} different functions, as there would be 16 distinct input vectors repeated in the fashion 1,2,4, . . . 2^{15} . However, the four outputs of BWRS would not cover all possible input vectors for $n > 4$. The other limitation of BWRS is that all its outputs do not switch at the same rate, as can be seen from the example given below for 2-bit BWRS

```

0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
0 1 1 0 0 0 0 1 1 1 1 1 1 1 1

```

To overcome these limitations we use equal-probability PRBS. As shown in Section 2.2, it is also impracticable to generate these sequences for $n > 6$, if we would like to distinguish all possible distinguishable functions using a value of 'a' given by eqn. 2.2.18. Other choices of 'a' also make hardware realization difficult. So that it has been decided, for the sake of simplicity, to use PRBS with $a = \frac{1}{2}$, which can be realized easily using a 16-bit MLS counter.

Thus the test pattern generator (TPG) operates in two phases. The IC under test is applied BWRS in the first phase and PRBS ($a = \frac{1}{2}$) in second phase. In order to test any 4-input combinational circuit (which can have 4-input pins as any 4-pins of a 16-pin IC) deterministically, the four outputs of BWRS A,B,C,D are distributed among 16-pins in such a way that that any of the ABCD can be selected at any pin of the IC.

3.2 INPUT GROUPS FOR SEQUENTIAL CIRCUITS

Typical inputs to sequential circuits are clock, asynchronous inputs, like preset, clear, load etc. and synchronous inputs like J,K,D, synchronous clear, synchronous load etc. Some times special purpose inputs like clock burst and up/dn or mode controls are also needed. Some of the synchronous inputs can be treated in the same way as the inputs to combinational circuits, and the outputs of TPG can be applied to them

Table 3.21 gives the list of signals available for selection at each pin, taking into account the possibilities of either positive logic or negative logic for each input. As the outputs of TPG in phase one are the four outputs of BWRS (A,B,C,D) and in phase two they are sixteen outputs of PRBS, the entries corresponding to TPG outputs will be in the form BWRS/PRBS.

Let $A_1, A_2 \dots A_{16}$ be the outputs of Asynchronous generator
(active low)

$\bar{A}_1, \bar{A}_2 \dots \bar{A}_{16}$ be the outputs of Asynchronous generator
(active high)

K,L, Z be the 16 outputs of PRBS

A,B,C,D be the 4-outputs of BWRS.

3.3 TYPICAL TEST CYCLE

The testing is performed in two or three steps, depending upon whether IC under test is a combinational circuit or a sequential circuit. The first step of testing sequential circuits is initialization.

Table 3.2.1 Pin numbers and signals available for selection

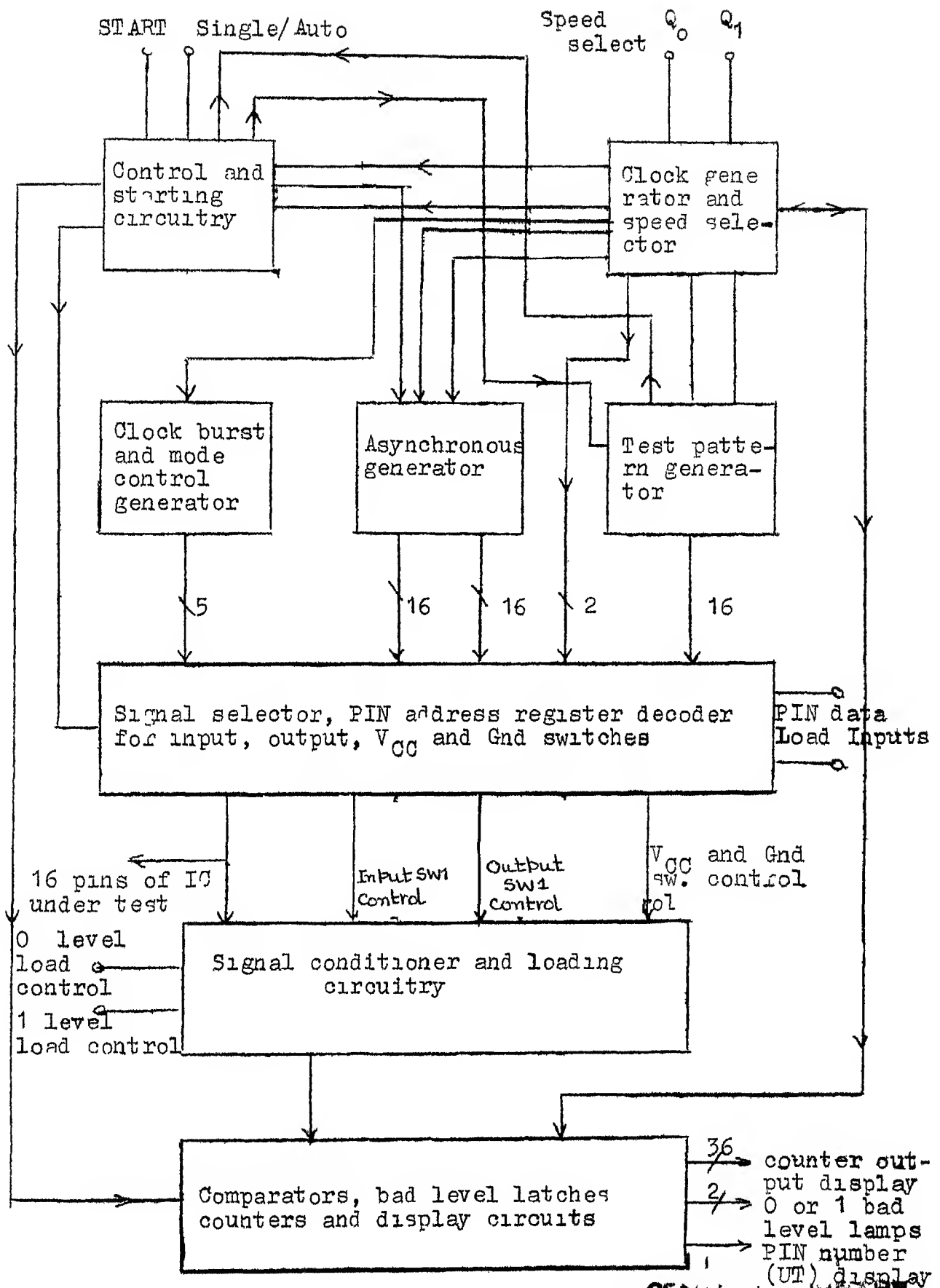
PIN Co No Code	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
0	For this code the pin becomes out-put																Sec Sect- ion 3.4
1	← Mode control →																See Sec 3.4
2	← Clock burst(negative logic) E																
3	← Clock burst(positive logic) F																
4	← Clock (positive logic) →																
5	$\overline{C/Y}$	$\overline{B/X}$	$\overline{A/W}$	$\overline{D/V}$	$\overline{D/Z}$	$\overline{A/K}$	$\overline{B/L}$	$\overline{C/M}$	$\overline{C/Q}$	$\overline{B/P}$	$\overline{A/O}$	$\overline{D/N}$	$\overline{D/R}$	$\overline{A/S}$	$\overline{D/T}$	$\overline{C/U}$	
6	A5	A6	A7	A8	A4	A3	A2	A1	A13	A14	A15	A16	A12	A11	A10	A9	
7	A5	A6	A7	A8	A4	A3	A2	A1	A13	A14	A15	A16	A12	A11	A10	A9	
8	$\overline{D/V}$	$\overline{C/U}$	$\overline{B/T}$	$\overline{A/S}$	$\overline{A/W}$	$\overline{B/X}$	$\overline{C/Y}$	$\overline{D/Z}$	$\overline{D/N}$	$\overline{C/M}$	$\overline{B/L}$	$\overline{A/K}$	$\overline{A/O}$	$\overline{B/P}$	$\overline{C/Q}$	$\overline{D/R}$	
9	$\overline{A/W}$	$\overline{D/V}$	$\overline{C/U}$	$\overline{B/T}$	$\overline{B/X}$	$\overline{C/Y}$	$\overline{D/Z}$	$\overline{A/K}$	$\overline{A/O}$	$\overline{D/N}$	$\overline{C/M}$	$\overline{B/L}$	$\overline{B/P}$	$\overline{C/Q}$	$\overline{D/R}$	$\overline{A/S}$	
10	$\overline{B/X}$	$\overline{A/W}$	$\overline{D/V}$	$\overline{C/U}$	$\overline{C/Y}$	$\overline{D/Z}$	$\overline{A/K}$	$\overline{B/L}$	$\overline{B/P}$	$\overline{A/O}$	$\overline{D/N}$	$\overline{C/M}$	$\overline{C/Q}$	$\overline{D/R}$	$\overline{A/S}$	$\overline{B/T}$	
11	← Clock (negative logic) →																
12	← Clock burst F' (positive logic) →																See Sec 3 4
13	← Clock burst E' (negative logic) →																See Sec. 3.4
14	← Not connected →																
15	← VCC/gnd →																

As the level signature obtained would depend very much on the initial state of the sequential circuit, we would like to start with the same initial state each time we test the identical circuit. Initialization is achieved by making use of preset, clear, load or mode control, inputs of the IC, whichever is available. In this step, which is performed in the beginning of each test cycle, each such pin gets active input (high or low) in turn. The pin which gets the last active input determines the initial state of the IC under test during this step logic inputs from TPG are kept low and '1's are not counted.

The following two steps are common to both combinational and sequential circuits. The IC under test is fed the appropriate input sequences selected by the user from Table 3.2.1, and number of ones are counted for each pin. The last step in the pinwise comparison of the count so obtained with the number of ones, obtained previously with good device of identical kind.

3.4 SYSTEM BLOCK DIAGRAM

The system block diagram is shown in Fig 3.1. The clock generator and speed selector generates and selects the frequency at which testing is performed (called test frequency). Pressing a start switch initiates the testing. The control circuitry gives the clear and load signals to various other circuits. It also gives control signals to help the initialization of sequential circuits. It incorporates the facility of



CENTRAL LIBRARY
Acc. No. A 637961

Fig. 3.1 System block diagram

22

testing either with a single-step or automatically with 16-step sequence TPG generates two phase test sequences as discussed in Section 3.1 Asy generates the signals, to be selected for Asynchronous preset, clear, load etc. inputs It can also be used for synchronous load, clear, mode control etc Clock burst generates four clock bursts, two with in active high level and two with inactive low level. (positive logic and negative logic) when one clock burst output is giving clock bursts the other remains at its inactive level and vice-versa. This circuit also generates mode control signal which goes high and low periodically, with period of 16 clock pulses.

Signal selector selects the input patterns from among 16 input choices available described in Table 3.2.1, depending upon the 4-bit address given for a particular pin. These addresses for all 16 pins are stored in 4 16-bit shift registers. These addresses are decoded to get the information whether a pin is input, output, $V_{CC}/\text{gnd.}$ or Not Connected (NC).

The signal conditioner translates the TTL levels to worst case input levels. The outputs of the ICs^{are} loaded according to the user choice and they are tested for bad levels, and one level is gated to the counter to obtain the level signature. At the end of test cycle the counter outputs are transferred to a shift register and the counter is cleared. The contents of the shift register are displayed for comparison with the previously obtained count for the particular pin number, which is also displayed.

.

CHAPTER 4

TEST SIGNAL GENERATOR AND CONTROL

This chapter deals with the design of the circuit needed for generation of the various test signals as well as for the control functions. The block diagram, the detailed circuit diagram, the actual IC lay out on the PCB, are given in Figs. 4.1, 4.2 and 4.3 respectively.

4.1 CLOCK GENERATOR AND SPEED SELECTOR

Sequential circuits have asynchronous inputs e.g. PRESET, CLEAR, MODE, etc., which must be held at the inactive level during the active edge of the clock. As the clock may either have positive logic or negative logic, the test pattern must ensure that the transitions in all the asynchronous inputs take place in between consecutive clock edges. This necessarily implies that the system clock must be chosen to have twice the frequency of the clock to the IC under test. Different logic families require different test frequencies, e.g., the maximum prescribed clock-rates for standard TTL family range from 20 to 50 MHz, while CMOS IC's can have clock-rates from 500 KHz to 5 MHz. Accordingly four choices have been allowed for test frequencies: 10 MHz, 5 MHz, 1 MHz, and 500 KHz. Correspondingly we need 20 MHz, 10 MHz and 1 MHz as the system clock frequencies. The system clock is generated using a 20 MHz crystal.

oscillator (G_{1-3}) and frequency dividers (FF_{1-4} , CTR_1) as shown in the circuit diagram. The system clock (SCK) and the test clock (TCK) are selected by a dual 4 to 1 multiplexer (ux1) from these two sets of four frequencies each.

4.2 TEST PATTERN GENERATOR (TPG)

As explained in Chapter 3, there are two phases of testing, the outputs of the TPG being those of the 4-bit BURS generator, in the first phase and those of the 16-bit PRBS generator in the second phase. The three circuit blocks generating the test patterns are discussed below

4.2.1 BURS Generator

The BUR sequence, as dictated by eqn. 2.1.6, will have to consist of vectors having magnitudes 0,1,2, ... 15 repeated 2^0 , 2^1 , ... 2^{15} times respectively. This can be implemented in a number of different ways, the chosen scheme generates the k th vector 2^k times consecutively, starting with $k=0$ and incrementing k after every set of 2^k clock pulses. While any scheme of realization would be equally applicable to combinational circuits, the chosen scheme permits the testing of sequential circuits in a properly cycled fashion.

The 16-bit down counter (CTR_{2-5}) is preset with the 16-outputs of the shift register (SR_{1-4}), which is preloaded with the bit sequence 000...1 at the beginning of each test cycle. The Ripple Carry of the down counter is the right shift clock

of the shift register, so that the magnitude of its contents becomes doubled every time the down counter reaches the zero count., and the down counter is also preset with this new value. In this manner the 16-bit down counter generates Ripple Carry output at the intervals of $2^0, 2^1, \dots 2^k \dots 2^{n-1}$ clock pulses. The 4-bit BWRS outputs are generated by the 4 outputs of a binary counter (CTR_6) clocked by this Ripple Carry

4.2.2 PRBS Generator

This is a simple 16-bit MLS counter (CTR_7) the feedback stages being 1,3, 12 and 16.

4 2.3 Pattern multiplexers

The 16 outputs of the PRBS generator and the 4-outputs of the BWRS generator are given to 16 2-to-1 multiplexers. (MUX_{2-5}), which are controlled by the 'Ø control' output of the control circuit.

4.3 ASYNCHRONOUS SIGNAL GENERATOR (ASG)

The requirement of the ASYNCHRONOUS inputs to any sequential circuit are (i) they should not become active too often, in order to allow the sequential circuit to complete its cycle length (ii) no two of them should become active simultaneously as within the same chip there could be RESET as well as CLEAR (iii) its transition edges should not coincide with that of the CLOCK (TCK) (iv) the choice should be given to change the transition edges of these signals while CLOCK is either at LOW or HIGH level.

In order to realize this the transition edges of ASG outputs are placed exactly in the middle of the two edges of the TCK using a J-K flip-flop (FF_5) with $J = TCK$ or \overline{TCK} and $CK = \overline{SCK}$, $K = 1$. The output of this flip-flops has its transition edges in between the two edges of TCK, and its positive edge occur while TCK is high if $J = \overline{TCK}$. This selection of J is done using an EXCLUSIVE or gate (G-4) and control bit 4.

The output of flip-flop (FF_5) is fed as clock to a binary counter (CTR_7). The Max-Min of this counter would occur after 16 clock periods and its transition edges would be placed as desired. This is shown in the waveforms given in the Fig. 4.3.

This Max/Min is given to the inputs of 16 2-input NAND gates (G_{5-20}). The other inputs of these NAND gates are the outputs of a 16-bit ring counter (SR_{5-8}) preloaded with 000 . . . 1. The right shift clock of this ring counter is the same Max/Min output. So that the outputs of NAND gates will go LOW after 256 TCK pulses for the duration of the Max/Min. These outputs are denoted by A_1, A_2, \dots, A_{16} are active low denoted by $\overline{A}_1, \dots, \overline{A}_{16}$ and are also made available G_{21-36} and they are denoted by $\overline{A}_1, \overline{A}_2, \dots, \overline{A}_{16}$ and are active HIGH type.

The D output of the counter (CTR_7) discussed above is given to a toggle flip-flop (FF_6). Thus the outputs of this flip-flop would remain HIGH or LOW for 16 TCK pulses and its transition edges would also be placed between the edges of TCK. This output is used as the Mode Control.

4.4 CLOCK BURSTS

Certain counters and shift registers have two CLOCK inputs only one of which is pulsed at a time with other remaining at its inactive level. While MODE control is changing its level both of the CLOCK inputs must remain at their inactive levels.

The Mode Control Signal discussed above is used to generate two pairs of clock bursts, to be used for two clock in an IC under test, one pair for negative - logic and the other for positive logic. It is very easy to see that mode \overline{TCK} and $\overline{Mode} \overline{TCK}$ give negative-logic Clock Bursts and $Mode + TCK$ and $\overline{Mode} + TCK$ give positive-logic Clock Bursts. (G_{37-40})

The four Clock Bursts and Mode Control are buffered and taken out (G_{41-45}).

4.5 CONTROL CIRCUIT

The control circuit has to perform the following tasks.

- (i) When power is switched ON the system should get reset automatically,
- (ii) when 'start test' signal is given control circuitry should initiate the testing and continue till end,
- (iii) when it receives 'End test' signal the instrument should be reset again,
- (iv) it should generate the signal which would indicate the phase of the test cycle under progress,
- (v) should generate various clearing and loading signals required by the rest of the circuitry.

To perform the above mentioned tasks the control circuit has four outputs and four inputs, the inputs being .

(i) Power ON, reset, (ii) 'Start test' signal, (iii) 'End of phase I' signal, (iv) 'End of phase II' signal, and the outputs being, (i) 'Test ON' output $\cdot C^*$, (ii) clear signal 1: CLR1, (iii) Clear signal 2 CLR2, (iv) phase indicator $\cdot \emptyset$ control.

As explained in Section 3.3, initialization cycle requires that each pin should receive an active Asynchronous signal from ASG once, the initialization cycle length would, therefore, be 256 TCK periods. At the beginning and at the end of this initialization cycle (INC) the ASG must be reset, whereas during INC, ASG should be enabled. The TPG should be reset at the beginning of the INC and it should be enabled only at the beginning of the actual test cycle. These requirements lead to the need for two different clear signals, one of them (CLR1) being the clear signal for the ASG and the other (CLR2) being the clear signal for the TPG.

These two clear signals are generated using flip-flops (FF₇₋₈), two binary counters (CTR₈₋₉) and some gates (G₄₆₋₅₂). As the transition edges of these clear signals should not coincide with that of TCK, the same technique, as discussed in Section 4.3 is used.

As long as 'Start Test' signal is not given CLR1 and CLR2 should remain LOW and also various other circuitry must remain

This is achieved using a NAND Latch (G_{53-54}) (output C^*) which is cleared when power switch is made ON. Pressing the switch $S1$ causes C^* to go high, thus initiating the testing.

Depending upon which phase of the test cycle is going on (\emptyset I or \emptyset II) an appropriate 'End of phase I/II' signal is generated. In the case of phase I the ripple carry of counter (CTR 6), NANDED with the \emptyset control signal (G_{55-57}) is given to the clock of a Toggle ^{flip-flop} (FF_9) which is initially cleared with C^* . And in the case of \emptyset II the output combination of all '1's is detected using two 8-input NAND gates (G_{58-59}) and then ^{both the outputs are} NORED (G_{60}). The output of the NOR gate is fed to J input J-K flip-flop (FF_{10}) with $ck = TCK$. Both the end of phase signals of ANDED with C^* (G_{61}) generates the 'End-Test Cycle' which is a negative going edge. This is used to trigger a monostable (mono 1). The output of mono 1 is utilized to transfer the contents of the '1's counter to the shift register (To be discussed in Chapter 6). It is also used to trigger another monostable whose output is used to clear the flip-flop (FF_{11}) and '1's counter. Mono 2 is also triggered when the power is switched on thus resetting the circuit. Clearing flip-flop (FF_{11}) again repeats the whole cycle. The phase control signal is generated by using flip-flops (FF_{11-12}) and a NAND gate (G_{62}).

CLR2 is given to the J-K flip-flop (FF_{13}), whose output goes high at the end of test cycle. This output is given to the counter (CTR_{10}). If CTR_{10} is preset with 0000 then the Max/Min would go HIGH after 32 CLR2 gates or 16 complete test cycles, or if CTR_{10} is preset with 1111, the Max/Min would go HIGH after one complete test cycle, thus giving the selection of single step or 16 steps automatically. The Max/Min of CTR_{10} is given to a J-K flip-flop (FF_{14}) which is initially cleared with C^* , the negative edge of Max/Min causes the output \bar{Q} of (FF_{14}) to go LOW. The \bar{Q} of FF_{14} is given to an AND gate (G_{63}) causing the NAND latch output C^* to go LOW. The NAND latch is cleared when power is switched on as the other input of the AND gate (G_{63}) is connected to V_{CC} through a resistor and to the gnd through capacitor. Once C^* goes LOW unless switch S_1 is pressed the next cycle will not begin.

4.6 DELAY COMPENSATION

The circuits used to generate different test signals uses counters, flip-flops, gates etc. which have different propagation delays, the overall delay, from the active edge of SCK to the final output transitions will be the sum of all individual delays. Moreover, different circuits will have different overall delays. In some cases the delay exceeds 50 nsec. If we use a system clock frequency of 20 MHz, the various edge relationships assumed in the design would be highly disturbed and may not even remain valid.

The delays of the different propagation channels have therefore been partially compensated by feeding variously delayed SCK and TCK pulses to different clock inputs of the TPG and ASG. While these clocks are fed undelayed to the control circuit (as it has maximum delay). In order to take this deliberately increased delays into account, the clock to the IC under test is also a delayed version of TCK.

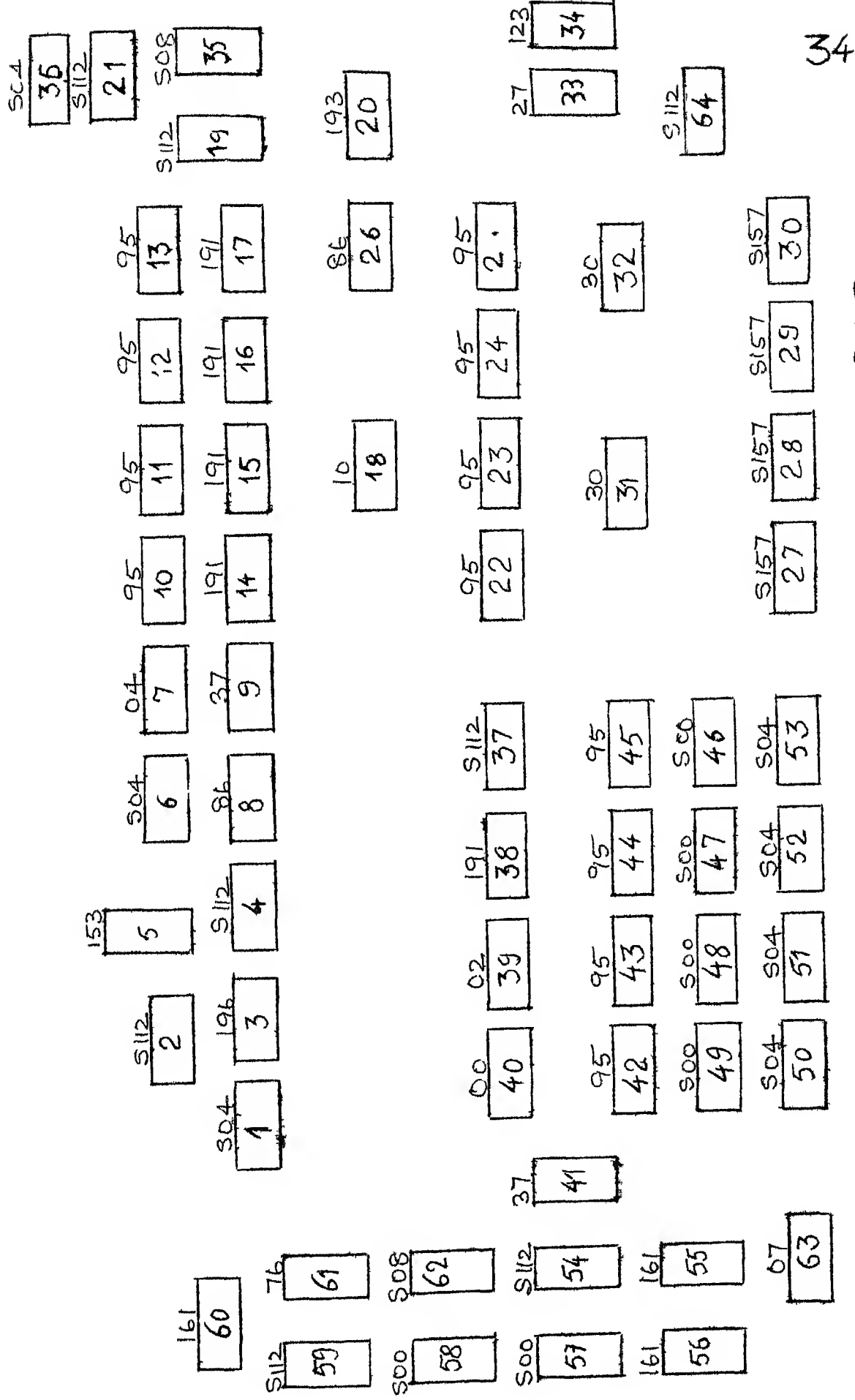


FIG NO 43 IC LAYOUT ON PCB

CHAPTER 5

INPUT SIGNAL SELECTION

This chapter deals with the design of the circuit needed for pin data storage, data loading, signal multiplexers and decoders for controlling various switches like input, output, V_{CC} or gnd switches. The block diagram, detailed circuit diagram and actual IC layout on the PCB are given in Fig 5.1, 5.2, 5.3 respectively.

5.1 PIN DATA REGISTER (PDR) AND DATA LOADING CIRCUITRY

Any pin of an IC can, in general, be classified in any of the four different ways: (i) input, (ii) output, (iii) V_{CC}/GND , and (iv) not connected. We have provided 3 choices of test signals, as described in Sec. 3.2. Thus in all 16 different possibilities exist for a pin and to specify which of these is to be selected for a particular pin, we need a 4-bit code for each pin (SR_{1-8}).

Pin data register 16 bit x 4 bit can store (SR_{1-8}) the information required for all the 16-pins of an IC under test. This information must be loaded into the PDR prior to the testing.

The four bit code is entered serially by pressing the appropriate push button switches (for '0' or for '1'). The serial data is first converted into parallel using three monostables (mono 1-3), a shift register (SR_9) and a gate (G_1). The outputs

of SR_9 are displayed using 4 LEDs. At the end of 4th bit, pressing the 'Center Data' switch (which triggers a monostable (mono-4), whose output is given to the clock of PDR) will load the four bits in the PDR.

5.2 SIGNAL MULTIPLEXERS

We need 16-to-1 multiplexer (mux 1-16) corresponding to the 16 possibilities discussed in Sec 5.1. The details of codes for various possibilities have been given in Table 3.2.1. Each multiplexer receives the 4-bit address from the PDR. The strobe of these multipliers is given CLR1, so that they are enabled during the INC (initialization cycle) also.

5.3 PIN DECODERS

The codes selected for the above mentioned 4 classifications of an IC pin are given below.

output	13 inputs	V_{CC}/GND	NC
0000	0001-1101	1110	1111

Corresponding to each pin there is an input switch and an output switch. A few pins also have V_{CC} and gnd switches. The inputs switches are to be switched OFF only if the pin is selected as V_{CC}/GND or as NC. The output switch is to be switched ON only, the pin is an output. And V_{CC}/GND switch is to be switched ON only if the pin is V_{CC} or GND. Then we have

$$\text{INS} = \text{input switch control} = \overline{Q_3 \cdot Q_2 \cdot Q_1}$$

$$\text{OUTS} = \text{output switch control} = \overline{Q_3 + Q_2 + Q_1 + Q_0}$$

$$V_{CCS} = \text{INS} + Q_0$$

$$\text{Gnds} = \overline{\text{INS} + Q_0}$$

The sixteen decoders for INPUT switch are 3-input NAND gates (G_{2-17}) The sixteen decoders for OUTPUT switches are 16 4-input NOR gates (G_{18-33}) and for V_{CC} /Gnd switches are 3 OR gates for V_{CC} and 4 NOR gates for Gnd. (G_{34-36} , G_{37-40})

V_{CC} and Gnd switches are not needed for every pin. Moreover, V_{CC} and Gnd switches are mutually exclusive for commercial 16 pin ICs. The pins used as V_{CC} and Gnd in commercial IC's (treating a 14-pin IC as an equivalent 16-pin, one with pin 8 and 9 not connected) have been found as follows

V_{CC} - pin 4,5 or 16

Gnd - pin 7,6,12 or 13.

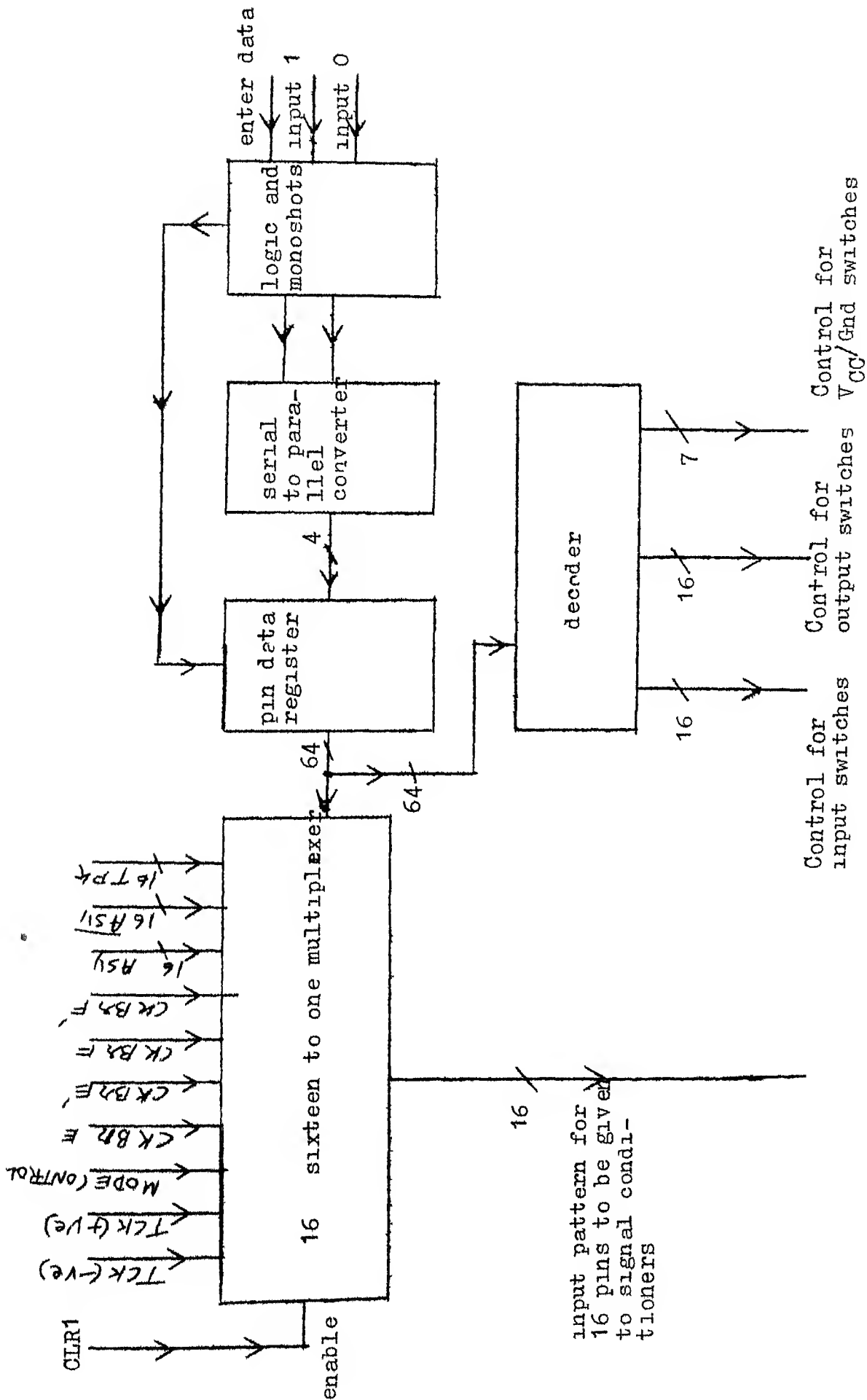
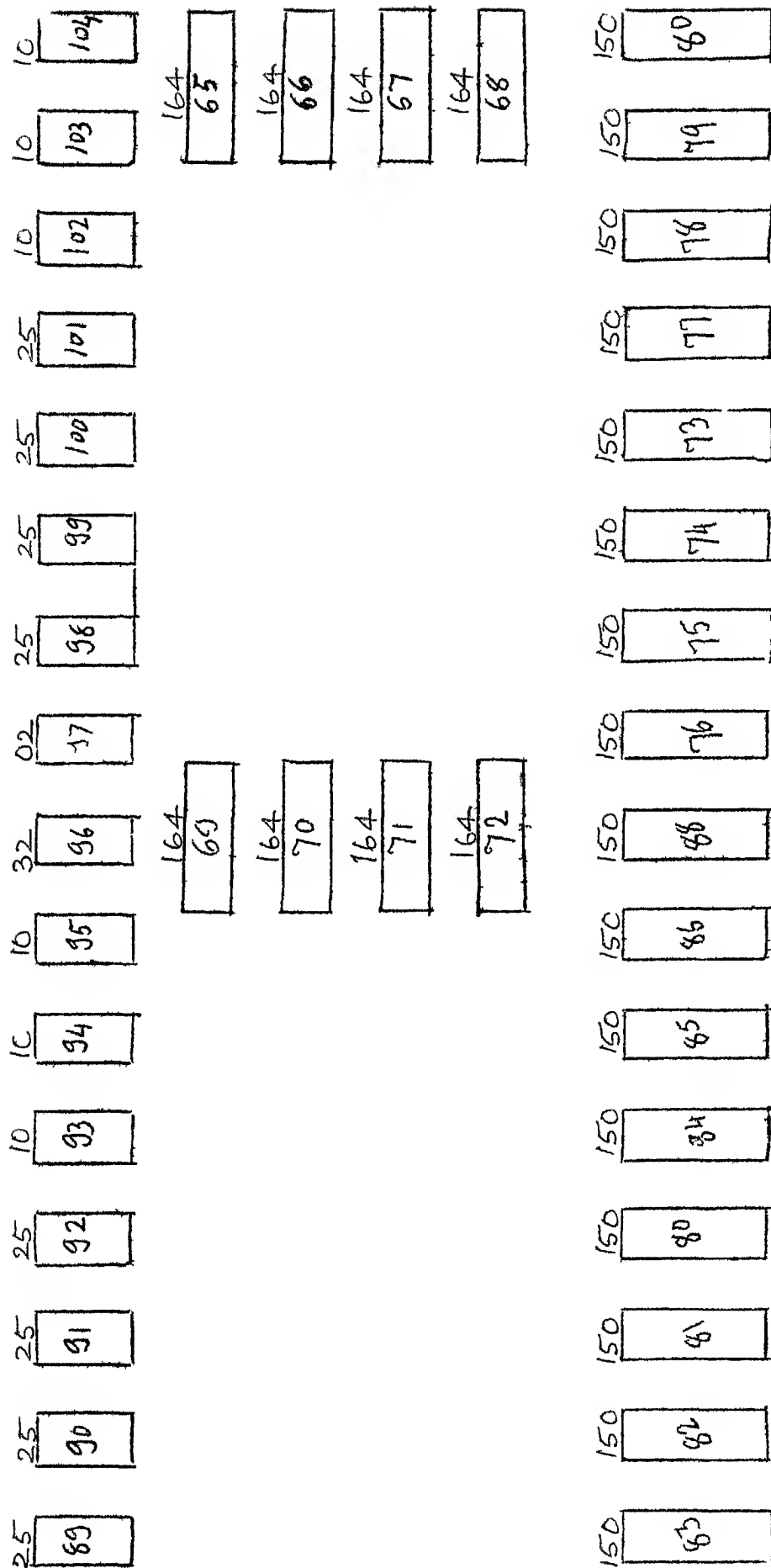


Fig. 5.1 Block diagram of signal selector pin address register and switch decoders



F16 5.3 IC Layout on PCB

CHAPTER 6

INTERFACE CIRCUIT AND SIGNATURE COUNTER

We discuss in this chapter the circuits involved in the application of the actual input signals as well as the checking of the actual outputs of the IC under test (ICUT). These circuits consist of the Interface circuit, the output (analog) multiplexer, Bad level detector Signature counter and display circuit. The block diagram of these circuits and their interconnections is shown in the Fig 6 1.

6.1 INTERFACE CIRCUIT

An Interface circuit is needed between the outputs of the signal multiplexers discussed in Sec. 5 2 (mux 1/73, mux 16/88) and the terminals of ICUT to take care of the following requirements .

- 1) the same pin can be either input, output, V_{CC}/Gnd or NC (no connection),
- 11) worst-case input levels should be applied, to the ICUT input terminals and (111) ICUT outputs are to be loaded according to the user's choice. One should be able to set the loading conditions for both the output levels (HIGH or LOW) independently and with reasonable accuracy, moreover one should be able to switch off the loading circuit when the pin is not selected as an output.

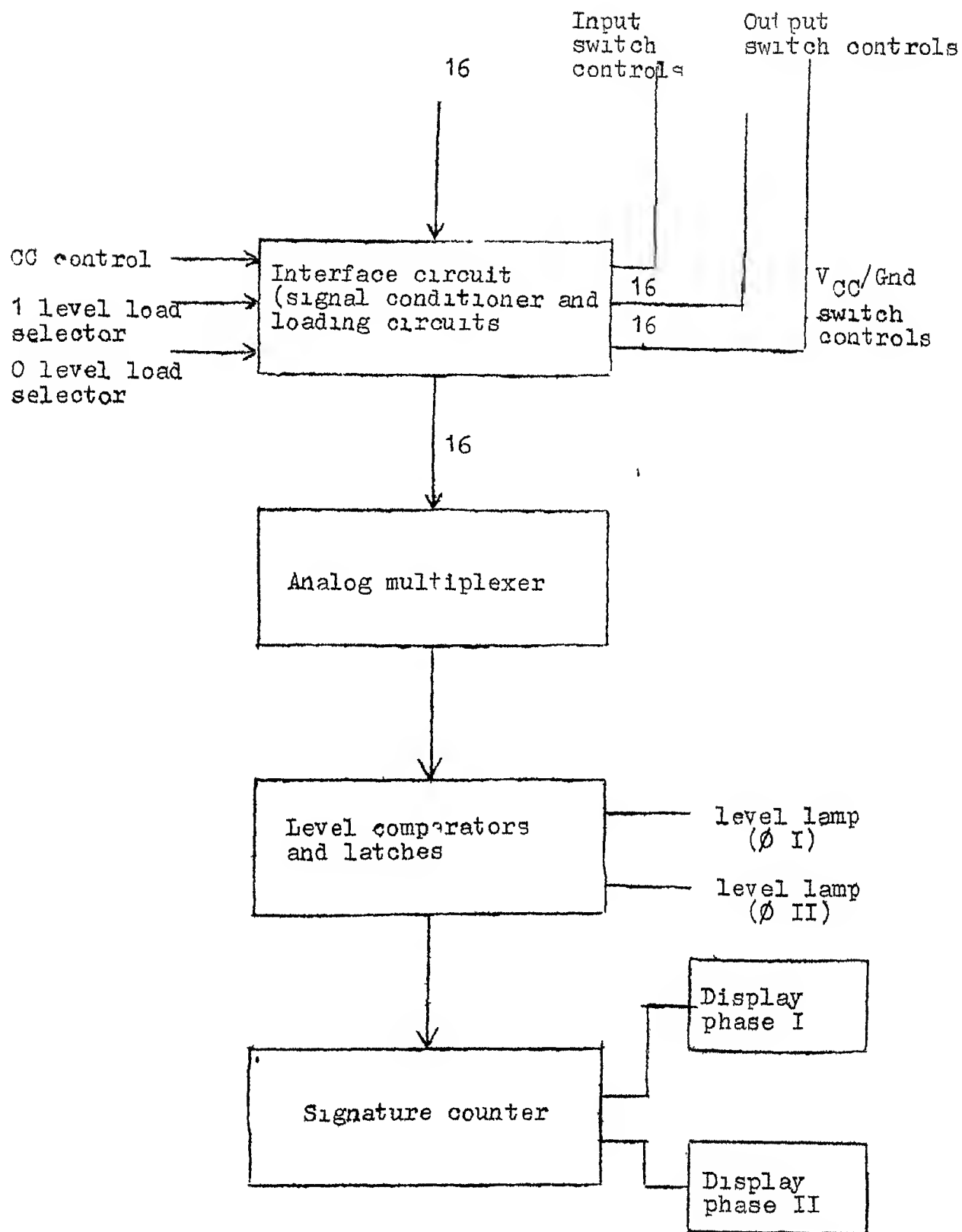


Fig. 6.1 Block diagram for signal conditioner loading circuitry and signature counter

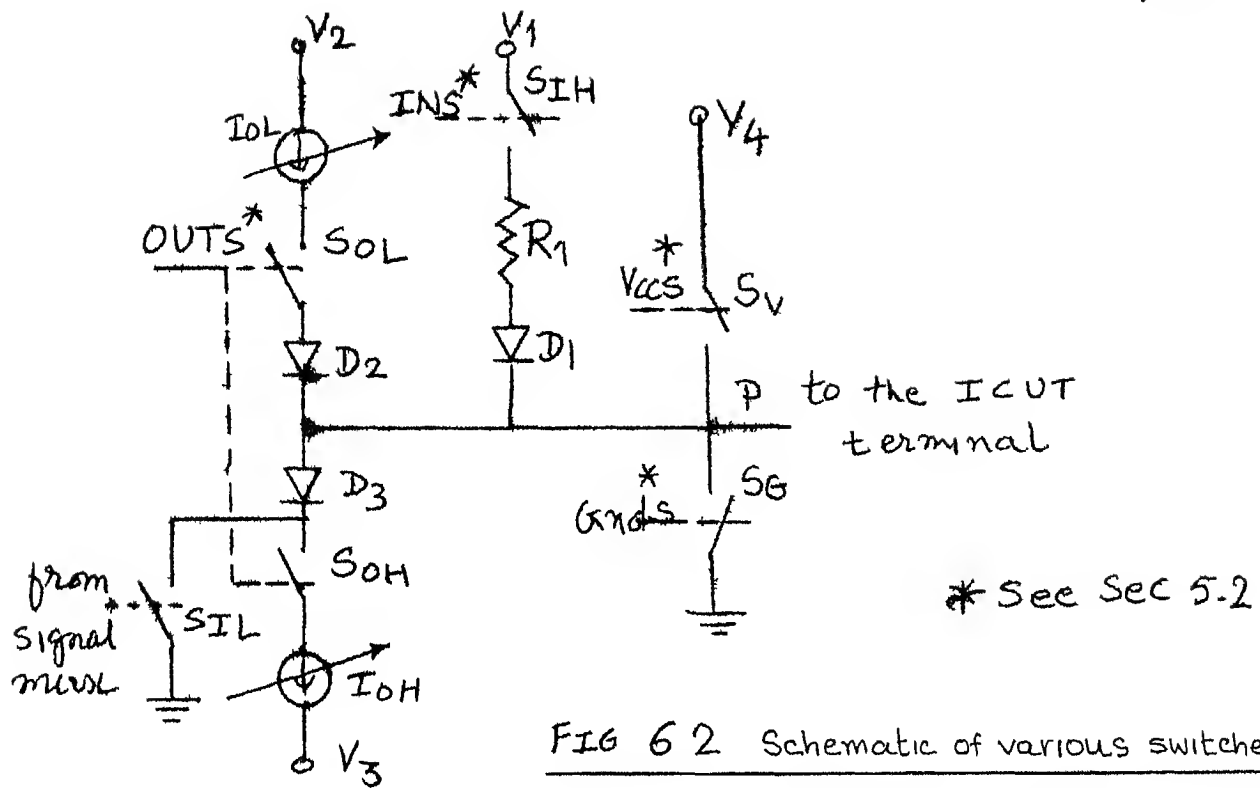


FIG 6 2 Schematic of various switches and

Loading circuitry

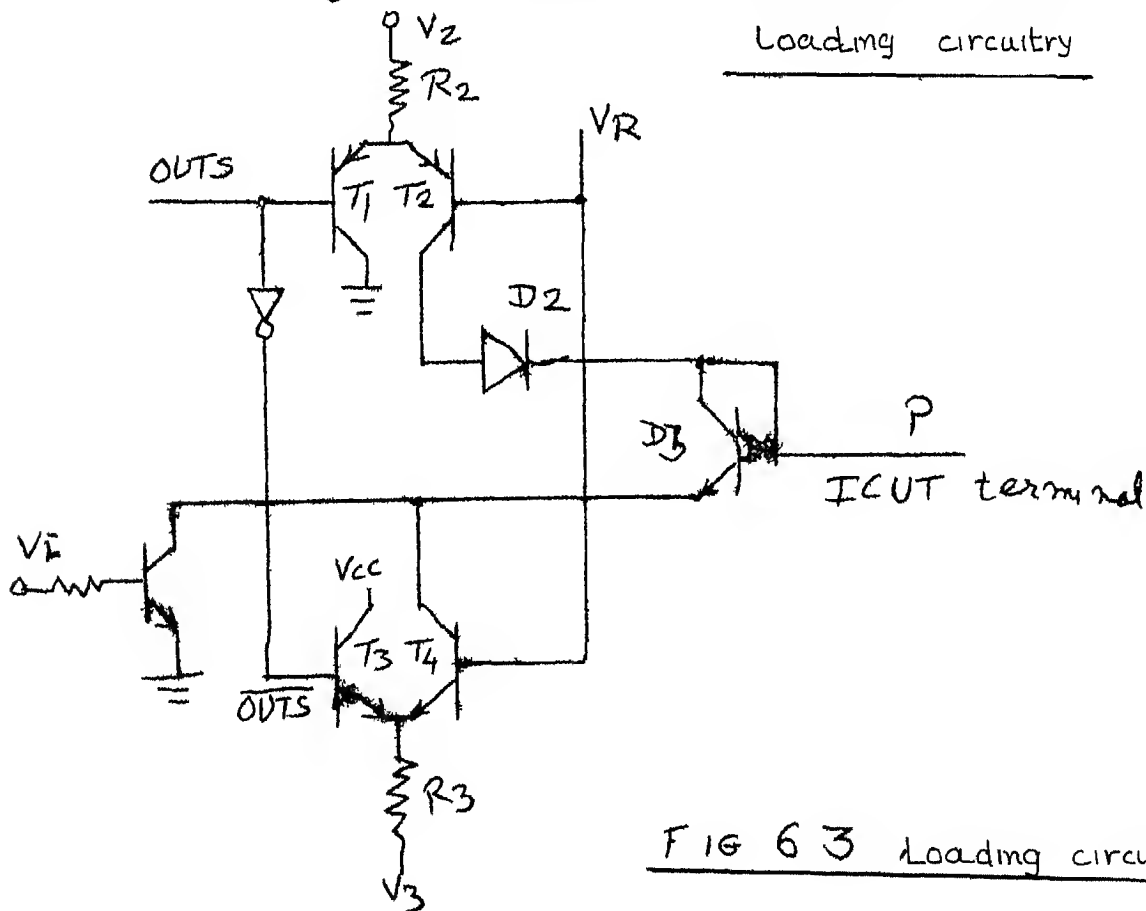


FIG 63 Loading circuit

The functional diagram of the interface circuit, used with each ICUT pin, meeting these requirements is shown in the Fig 6.2.

Tables 6 1 gives the required switch positions and resulting voltages at the ICUT pin for different connections S_{IL} represents the level of the input v_1 applied from the signal multiplexer (S_{IL} OFF for LOW input and ON for HIGH input).

Table 6.1.1 Interface switch positions

Desired connection	S_o	S_{IH}	S_v	S_G	V_p		S_{IL}	OFF
					S_{IL}	ON		
NC	OFF	OFF	OFF	OFF	-		open	
output	ON	ON	OFF	OFF	-		v_o	
input	OFF	ON	OFF	OFF	$V_s^+ + V_{d3}$		$V_1 - V_{D1}^+$	
V_{CC}	OFF	OFF	ON	OFF	-		$V_4 - V_s$	
Gnd	OFF	OFF	OFF	ON	-		V_s^+	

⁺ $V_s = V_{CE \text{ sat}}$ as these switches are implemented using transistor^s

The resistor R_1 acts as a current limiter if v_1 is HIGH, S_{IH} is kept ON even if the pin is selected as an output, in order to provide a pull up in case an open-collector of the ICUT is under test. The diode D_1 has been added to prevent R_1 from loading a totem-pole output (as $V_1 < V_{CC}$).

When the pin is connected as an output, the loading levels for '0' and '1' states of the ICUT output are controlled by two current sources I_{OL} and I_{OH} . D_2 and D_3 are used to make these loadings mutually independent.

The switches S_V and S_G are implemented using a PNP transistor and an NPN transistor. (Q_{1-7}) respectively. The two current sources I_{OL} & I_{OH} along with the switches S_{OL} & S_{OH} are realized using two differential pairs of PNP and NPN transistors (Q_{8-71}) respectively. Open collector inverter gates (G_{1-16}) are used for switches S_{IL} . Analog gates are used for the switch S_{IH} (AS_{1-16}) as the current required through this switch is not high enough to call for a transistor switch.

As explained in the Sec. 5.3 the V_{CC} and Gnd switches are not needed for every pin. The Gnd terminal^{of} the ICUT is above the system ground by V_S , which ranges from .1 to .3 V.

Thus when v_1 is LOW the ICUT pin is at, a voltage $V_p - V_S = V_1 - V_{D1} - V_S$. With respect to the TTL Gnd , and hence the worst-case V_{IH} can be applied by adjusting V_1 ($V_1 = 2.6$ V for $V_{IH} = 2.0$ V)

The two differential pair connection for output loading are shown in Fig. 6.3.

When the ICUT pin is selected as an output, transistors T_1 and T_3 are off and all the current through R_2 and R_3 flows through transistor T_2 and T_4 . If V_p is low the load current sunk in the ICUT terminal is given by

$$I_{OL} = (V_2 - V_R - V_{BE}) / R_2, \quad (6.1.1)$$

if V_p is HIGH the load current sourced by the ICUT terminal is given by

$$I_{OH} = (V_R - V_3 - V_{BE}) / R_3 \quad (6.1.2)$$

I_{OL} and I_{OH} can be varied by varying V_2 and V_3 .

If the pin is not selected as an output, $\overline{OUTS} = \text{HIGH}$ ($\overline{OUTS} = 0$, obtained using inverters (G_{17-32})) T_1 and T_3 conduct so as to divert the current from T_2 and T_4 which makes them off. V_R is selected somewhere near TTL threshold. (≈ 1.2 V).

6.2 ANALOG MULTIPLEXER AND BAD LEVEL DETECTION

Any TTL output is guaranteed to have a minimum value of 2.4V for HIGH level, and maximum value of 0.4 V for low level. Any output voltage lying in the range 0.4V - 2.4V is therefore considered as bad level for TTL. This has to be detected by means of a window comparator. Clearly the hardware will be prohibitively expensive if such a fast window comparators is used for every pin of ICUT. The hardware can be considerably minimized if we perform bad level detection for one pin of a time.

This is accomplished using a 16 - to - 1 analog multiplexer. Two analog multiplexers (mux_{1-2}) are used along with a binary counter (CTR_1) and an inverter (G_{33}), whose outputs are used

as the addresses of the mux₁₋₂. The outputs of the counter will indicate the pin to be selected for bad level detection. CTR₁ is clocked with the Q output of the flip-flop FF_{13/61}, whose output goes high at the end of each test cycle.

The output of the analog multiplexer is given to three comparators (comp₁₋₃) with comparison levels of 1.6V, 2.6V, .6V (taking in to account the non zero values of V_S).

The comparator comp₁, with comparison level 1.6V, serves as a high speed buffer with very little loading. The output of the other comparators comp₂, comp₃ are interconnected using the internal strobing circuit, to produce an output BLG which would go LOW if a bad level is detected. But this output alone can not be directly utilized for bad level detection due to following reasons.

- (i) this output will go low, indicating a bad level, during the transition period of any ICUT output.
- (ii) If the pin is selected as an input the signal levels will be always bad owing to the choice of worst-case input levels.
- (iii) As explained earlier the voltage source V_1 , through the transistor switch S_{IH} , acts as the pull up for an open-collector ICUT output, so that the HIGH level output of an open collector will always be registered as at bad level, which is not true.

To take care of the above mentioned situations, the final bad level indication BLI is generated by means of the following logic.

$$BLI = BLC * TD + \overline{OUTM} + COMP_1 * OC ,$$

where TD is the transition duration, generated by a monostable (mono₁) triggered with the positive edge of the undelayed TCK (2y output of the mux_{1/5}).

OUTM = 1 if the pin under test is an output.

and OC = 1 indicates open-collector output.

TD is adjustable over the range 60-400nsec, to take care of various delays. The signal \overline{OUTM} is generated using the counter (CTR₁) and a 16-to-1 multiplexer (mux₃). The inputs to the mux₃ are the output switch controls for all 16 pins of the ICUT.

The final OR gate (G₃₆) output is given to the bad level latch. Any time the output of G₃₆ goes LOW, the flip-flop (FF₁) is set to '1' permanently till the end of the test cycle.

6.3 SIGNATURE COUNTER AND DISPLAY

The negative edge of the output of mono₁ triggers another monostable (mono₂). The output Q mono₂ of this monostable is used alongwith the signals comp₁ and BLI to generate the clock SCCK to the signature counter as follows,

$$SCCK = Q_{M2} * COMP_1 * \overline{BLI}.$$

The signature counter consists of synchronously cascaded counters (CTR₂₋₅). The 16-bit output of this counter is transferred parallelly to a 16-bit shift register (SR₁₋₄) at the end of each phase, clocked by the \bar{Q} output of the mono 1/34 (sec 4.5). The output of this shift register is also transferred to another 16-bit register (SR₅₋₈) by the same clock.

Thus at the end of a complete test cycle SR₁₋₄ will store the signature for phase II and SR₅₋₈ will store the signature for phase I. Two D flip-flops (FF₂₋₃) are used to store the information about the bad levels in a similar manner.

At the end of each phase of testing the \bar{Q} output of mono 2/34 (see Sec 4.5) clears the signature counter (CTR₂₋₅) and bad level latches.

Signature for each phase, stored in (SR₁₋₄, SR₅₋₈) are displayed in octal using 12 seven segment displays (Display 1-12, drives 1-12) the bad levels for each phase are also displayed using two LEDs. The number of the pin under test is also displayed (display 13, driver 13)

The detailed circuit diagram is shown in Fig. 6.4.

CHAPTER 7

CONCLUSION

The operating instructions and suggestions for possible future developments are discussed in this chapter.

7.1 OPERATING INSTRUCTIONS

The test procedure for an IC with unknown signature consists of the following steps.

1. Select the code for each pin.
2. Load the codes, starting with pin 16 and proceeding down to pin 1, by first entering the code through the binary switches marked '0' and '1' and then by pressing the 'Enter Code' push button switch.
3. Set the test clock frequency.
4. Specify O.C. or totempole output
5. Choose the value of transition delay
6. Select the load levels
7. Obtain two level signatures for each pin one at a time for guaranteed specimen of ICUT, by pressing the push button switch marked 'Start Test' and note them down. (This step is eliminated if the signature is known for the same codes as in step (1)).
8. Obtain the signature for each pin for ICUT and compare with the previously recorded results for guaranteed device.
9. If ICUT fails the test check the bad level indicator to identify whether the fault is due to a bad level or a logical fault and test it again with lesser speed and/or lesser load.

7.1.1 Selection of Codes for Combinational Circuits.

- (1) If ICUT is 14-pin IC. renumber the pins so that it can be treated as 16-pin IC with pins 8 and 9 not connected
- (ii) Classify each pin in to four groups (1) output,
(2) V_{CC}/Gnd (3) NC (4) Input.
- (iii) Refer Table 3.2.1 to get the codes for the pins based on the above classification. For inputs try to assign the codes to the inputs of same logic element in such a way that A,B,C,D are repeated for least number of times, and K,L Z are never repeated.
- (iv) Tabulate the pin number, classification, input signal selected and code, starting with pin number 16.

This procedure is illustrated in Table 7.1.1 for 4 input NAND gate.

Table 7.1.1 Codes and Test Results of 4 input NAND Gate
(14 pin IC)

Pin No.	Pin Classification	Input Selected	Code	Count Phase I	Count Phase II
16	V _{CC}	-	1111		
15	Input	A/S	1010		
14	Input	B/P	1000		
13	N.C	-	1110		
12	Input	C/M	1010		
11	Input	D/N	1010		
10	Output	-	0000		
9	N.C.	-	1110		
8	N.C.	-	1110		
7	Gnd	-	1111		
6	Output	-	0000		
5	Input	A/W	1000		
4	Input	B/T	1001		
3	N.C.	-	1110		
2	Input	C/U	1000		
1	Input	D/V	1000		

7.1.2 Selection of Codes for Sequential Circuits.

Here the input pins are to be classified further in to following groups.

- (i) Clock +ve logic (ii) Clock -ve logic.

- (iii) Asynchronous inputs with active low
- (iv) Asynchronous inputs with active high
- (v) Mode control inputs
- (vi) If the chip has two clock inputs they should be given clock bursts. Classify here again for +ve logic and negative logic.
- (vii) Finally synchronous inputs like J,K,D,T, data inputs etc. are to be treated as inputs for combinational circuits.

After classifying the pins write pin numbers and codes in a tabular fashion, in the same way as table 7.1.1.

The procedure for code selection for sequential circuit is illustrated in Table 7.1.2 for the counter 74191.

Table 7.1.2 Codes and Test Results of 74191 Counter

Pin No.	Pin Classification	Input Selected	Code	Count Phase I	Count Phase II
16	V _{CC}	-	1111		
15	Input	A/S	1010		
14	Clock +ve	Ck +ve	0100		
13	Output	-	0000		
12	Output	-	0000		
11	Asy active low	A ₁₅	0110		
10	Input	B/P	0101		
9	Input	C/Q	0101		
8	Gnd	-	1111		
7	Output	-	0000		
6	Output	-	0000		
5	mode control	mode	0001		
4	Asy active high	\overline{A}_8	0111		
3	Output	-	0000		
2	Output	-	0000		
1	Input	D/N	1000		

7.2 SUGGESTED FUTURE DEVELOPMENT

The test procedure is lengthy, and there are many chances of making errors. Thus some form of automation is needed.

This can be achieved using a micro computer.

Data for some very common ICs can be stored in EPROMS, and for testing one needs to feed only the type number of the ICUT, which can be appropriately decoded, and corresponding codes and signatures can be fetched. Testing for all the pins can be performed in a continuous manner. The output of the mono 2/34 can be used to interrupt the micro computer and the signature counter outputs can be stored in the RAM by the computer and then compared automatically with the known signatures residing in the memory. Load conditions, speed selection, etc. can be either programmed or set manually.

BIBLIOGRAPHY

- 1 J P. Poage and E J McClusky, "Derivation of optimal test sequences for sequential machines', in Proc. Fifth Annu. Symp. Switching Theory and Logic Design, pp. 121-132, 1964, (Ref 1).
- 2 Jacques Losq, "Efficiency of random compact testing', IEEE Trans. on Computers, Vol C-27, No.6, June 1978, pp. 516-525 (Ref. 2).
- 3 J P. Roth, W.C Bouricius and R.P Schneider, "Programmed algorithm to compute tests to detect and distinguish between failures in logic circuits", IEEE Trans. Electron. Comput. Vol. EC-16, pp. 567-580, Oct. 1967.
4. W.G. Bouricius, E.P. Hsieh, G R. Putzolu, J.P. Roth, P.R Schneider and C.J. Tan, "Algorithms for detection of faults in logic circuits", IEEE Trans. Comput. Vol. C-20, pp. 1258-1264, Nov. 1971.
5. M.A. Breuer, "A random and an algorithmic technique for fault detection test generation for sequential circuits", IEEE Trans Comp. Vol. C-20, pp. 1364-1370, Nov. 1971.
6. J. Losq, "Referenceless random testing", Proc. 6th International Symposium on fault tolerant computing , pp. 108-113, June, 1976.

APPENDIX I

If a_j 's are the probabilities that j^{th} input would be 1, then the probability that an input vector would be A_k is given by

$$p_k = \prod_{j=1}^{n-1} f_j \quad (\text{A.1})$$

using eqn. 2.2.5. where f_j is defined as follows

$$f_j = a_j \quad \text{if } k_j = 1 \quad (\text{A.2})$$

$$f_j = (1 - a_j) \quad \text{if } k_j = 0$$

Thus for $k = 0$

$$P_0 = (1-a_0)(1-a_1) \dots (1-a_{n-1}) \quad (\text{A.3})$$

if $k=2^j$

$$P_{2^j} = (1 - a_0)(1 - a_1) \dots (1 - a_{j-1})a_j(1 - a_{j+1})(1 - a_{n-1}) \quad (\text{A.4})$$

using the eqns (A.3) and (A.4) and eqn. (2.2.2)

$$\text{we get } a_j = \frac{1}{2^{2^j} + 1} \quad j = 0, 1 \dots n-1 \quad (\text{A.5})$$

to show that this solution for a_j indeed satisfies (2.2.2), we expand k in terms of its binary representation.

$$k = \sum_{j=0}^{n-1} k_j 2^j \quad (\text{A.6})$$

using eqns. A.1, A.2, A.5, and A.6 we have

$$f_j = \frac{2^{2^j} (1 - k_j)}{2^{2^j} + 1} \quad (\text{A.7})$$

$$\begin{aligned} \therefore P_k &= \frac{\prod_{j=0}^{n-1} 2^{(2^j - (1 - k_j))}}{\prod_{j=0}^{n-1} (2^{2^j} + 1)} \\ &= \frac{2^{\sum_{j=0}^{n-1} (2^j - (1 - k_j))}}{2^N - 1} \\ &= \frac{2^N - 1 - k}{2^N - 1} \quad (\text{A.8}) \end{aligned}$$

thus, for the complementary situation,

$$\begin{aligned} a_j &= \frac{1}{2^{2^j} + 1} \\ p_k &= \frac{2^k}{2^N - 1} \end{aligned}$$

EE-1980-M-SHA-SIG